Amendments to the Claims:

This listing of claims replaces all prior versions and listings of claims in the application.

Listing of Claims:

1. (Currently Amended) A method for controlling concurrency of access to data in a database system, the method comprising:

~~(a)~~ partitioning a table in the database system into a plurality of partitions;

~~(b)~~ receiving a lock request for access to data in the database system, the lock request being a request for a page lock or a row lock for a corresponding row or page in the database system containing the data;

~~(c)~~ ~~determining~~ identifying a partition of the plurality of partitions that contains the row or the page in the database system containing the data;

associating the lock request with a partition lock on the partition that contains the row or the page in the database system containing the data; and

accessing the data using the partition lock.

~~(d) determining if the data has been committed; and~~

~~(e) if so, avoiding locking the partition in response to the request.~~

2. (Currently Amended) The method of claim 1, further comprising:

responsive to the data being committed at a time of receiving the lock request, accessing the data without using the partition lock.

~~(f) if it cannot be determined whether the data has been committed:~~

~~(f1) locking the partition in response to the request, and~~

~~(f2) granting access to the partition.~~

3. (Currently Amended) The method of claim ~~2~~ 1, wherein ~~granting access to the partition~~ accessing the data using the partition lock includes serializing access to the partition at a lock state to protect against interference in the form of updates to the partition.

4. (Currently Amended) The method of claim 3, wherein ~~locking the partition includes locking the partition at the lock state and~~ serializing access to the partition includes permitting lock requests access to the partition ~~for access requests~~ that are compatible with the lock state.

5. (Currently Amended) The method of claim [[4]] 1, wherein ~~the access to~~ accessing the data using the partition lock comprises ~~access by a plurality of applications~~ an application accessing the data through a single database system.

6. (Currently Amended) The method of claim ~~2~~ 1, wherein ~~the access to~~ accessing the data using the partition lock comprises ~~access by a plurality of~~ a second database system in a data-sharing environment accessing the data.

7-8. (Cancelled)

9. (Currently Amended) The method of claim ~~8~~ 1, wherein the lock request is a request for a shared lock.

10. (Currently Amended) The method of claim ~~8~~ 1, wherein the lock request is a request for an exclusive lock.

11. (Currently Amended) The method of claim 4, further ~~including~~ comprising:

receiving a lockmax value;

accumulating for an application, a number of lock requests for access to the ~~records in the table~~ data in the database system by the application;

comparing the number of lock requests with the lockmax value; and

when the number of lock requests equals the lockmax value, escalating the lock state.

12. (Currently Amended) A database management system implemented in a digital computer system, the database management system configured to ~~that manages~~ manage access to data in a database system ~~containing data in tables~~, the database management system comprising:

a database system component to partition a table in the database system into a plurality of partitions; and

a data manager ~~that;~~ configured to:

receive ~~receives~~ a lock request for access to data in the database system, the lock request being a request for a page lock or a row lock for a corresponding row or page in the database system containing the data;

identify a partition of the plurality of partitions that contains the row or the page in the database system containing the data;

associate the lock request with a partition lock on the partition that contains the row or the page in the database system containing the data; and

access the data using the partition lock.

~~determines if the data has been committed, and~~

~~if so, avoids locking the partition in response to the request.~~

13. (Currently Amended) The database management system of claim 12, wherein the data manager is further configured to access the data without using the partition lock responsive to the data being committed at a time the data manager received the lock request.

~~obtains a lock on the partition in response to the request, if it cannot be determined whether the data has been committed.~~

14. (Currently Amended) The database management system of claim ~~13~~ 12, further comprising a lock manager ~~that: grants the lock on the partition; and serializes~~ configured to grant serialized access to the partition at a lock state to protect against interference in the form of updates to the partition.

15. (Currently Amended) The database management system of claim 14, wherein ~~the database manager obtains the lock at the lock state and~~ the lock manager serializes access to the partition by granting lock requests ~~for locks~~ on the partition that are compatible with the lock state.

16. (Currently Amended) The database management system of claim 14, wherein the lock manager is coupled to a plurality of database systems of a data-sharing environment.

17. (Currently Amended) The database management system of claim ~~15~~ 12, wherein the ~~requests for locks represent requests~~ lock request is ~~for access to the table~~ from ~~a plurality of applications~~ an application coupled to the database management system.

18. (Currently Amended) The database management system of claim 15, ~~further including a lockmax value,~~ wherein the database manager is further configured to:

accumulate ~~accumulates~~ for an application a number of lock requests for access to ~~records in the table~~ the data in the database system by the application;

compare ~~compares~~ the number of lock requests with ~~the lock max~~ a pre-determined value; and

when the number of lock requests equals the ~~lockmax~~ pre-determined value, ~~requests~~ request escalation of the lock state.

19-20. (Cancelled)

21. (Currently Amended) The database management system of claim ~~20~~ 12, wherein the lock request is a request for a shared lock.

22. (Currently Amended) The database management system of claim ~~20~~ 12, wherein the lock request is a request for an exclusive lock.

23. (Currently Amended) A computer readable medium encoded with a computer program ~~instructions~~ for controlling concurrency of access to data in a database system, the computer program comprising computer executable instructions for:

~~(a)~~ partitioning a table in the database system into a plurality of partitions;

~~(b)~~ receiving a lock request for access to data ~~in a partition in the table~~ in the database system, the lock request being a request for a page lock or a row lock for a corresponding row or page in the database system containing the data;

identifying a partition of the plurality of partitions that contains the row or the page in the database system containing the data;

associating the lock request with a partition lock on the partition that contains the row or the page in the database system containing the data; and

accessing the data using the partition lock.

~~(c) determining if the data has been committed; and~~

~~(d) if so, avoiding locking the partition in response to the request.~~

24. (Currently Amended) The computer readable medium of claim 23, wherein the computer program further ~~comprising~~ comprises computer executable instructions for:

accessing the data without using the partition lock responsive to the data being committed at a time of receiving the lock request.

~~(e) if the data has not been committed:~~

~~(e1) locking the partition in response to the request, and~~

~~(e2) granting access to the partition.~~

25. (Currently Amended) The computer readable medium of claim ~~24~~ 23, wherein ~~granting access to~~ the computer executable instructions for accessing the data using the partition lock ~~includes~~ include computer executable instructions for serializing access to the partition at a lock state to protect against interference in the form of updates to the partition.

26. (Currently Amended) The computer readable medium of claim 25, wherein ~~locking the partition includes locking the partition at the lock state and~~ the computer executable instructions for serializing access to the partition ~~includes~~ include computer executable instructions for

permitting lock requests access to the partition ~~for access requests~~ that are compatible with the lock state.

27. (Currently Amended) The computer readable medium of claim ~~26~~ 23, wherein the computer executable instructions for accessing ~~access to~~ the data using the partition lock include computer executable instructions for having ~~comprises access by a plurality of applications~~ an application access the data through a single database system.

28. (Currently Amended) The computer readable medium of claim ~~25~~ 23, wherein the computer executable instructions for accessing ~~access to~~ the data using the partition lock include computer executable instructions for having a second ~~comprises access by~~ a ~~plurality of~~ database system in a data-sharing environment access the data.

29-30. (Cancelled)

31. (Currently Amended) The computer readable medium of claim ~~30~~ 23, wherein the lock request is a request for a shared lock.

32. (Currently Amended) The computer readable medium of claim ~~30~~ 23, wherein the lock request is a request for an exclusive lock.

33. (Currently Amended) The computer readable medium of claim 26, wherein the computer program further ~~including~~ includes computer executable instructions for:

receiving a lockmax value;

accumulating for an application, a number of lock requests for access to the ~~records in the table~~ data in the database system by the application;

comparing the number of lock requests with the lockmax value; and

when the number of lock requests equals the lockmax value, escalating the lock state.